

## DELIVERY OF SEQUENTIAL INFORMATION

5

### Technical Field

The present invention relates generally to electronic publishing. More particularly, the present invention relates to accessing sequentially issued information via an electronic network.

10

### Background of the Invention

15

The tremendous growth of the number of users accessing the Internet has resulted in an increase in the amount of information published on the World Wide Web. News, weather, and e-commerce, even entire novels are available for downloading over the Internet.

20

Content prepared for subscription delivery is typically periodical. In other words, the information is delivered only within certain "dated content" constraints. The recipient typically retrieves this content from a fixed Uniform Resource Locator (URL). In order to keep the information current, the publisher has to continually monitor the content during the delivery window. As used herein, the term "publisher" is used to mean the entity that makes content or information generally available to multiple persons, and can include an author of the content.

25

The timing requirements of this method of electronic publishing put significant demands on the publisher. The publisher must have the content ready before its window of availability. Additionally, the customer is forced to accept delivery of each issue during that window of time. For example, if CNN's Web site runs a series on the space program, each edition of the series must be ready prior to

being posted and the customer has a limited time in which to read it before it is replaced with the next in the series.

An additional problem faced by electronic publishers is that many publications are sequential in nature but do not require delivery during a specific time window. For example, multi-step tutorials or the chapters of a novel should be read in a sequential order but don't necessarily require that they be published within any predetermined time.

A problem faced by electronic publishing users is that every person has a different schedule when it comes to reading for pleasure. One reader may enjoy and have the time for reading daily while another reader may only have time to read a novel on a weekly basis. Current publishing and delivery systems cannot accommodate the delivery preferences of both types of readers subscribing to the same content. There is a resulting unforeseen need to provide electronic publishing of content having a sequential nature in a manner that allows a variable schedule delivery satisfactory to the consumer but without violating constraints placed on the sequentiality or timing of the content by the publisher or author for artistic, economic, or other reasons

### Summary of the Invention

The present invention encompasses the delivery of serialized content to a user's content receiving device by obtaining content having a plurality of portions arranged in a predetermined sequential order. A first delivery rule is received for the content and a next portion of the plurality of portions is determined for the user to receive in accordance with the predetermined sequential order. A portion of the plurality of portions of content is delivered to the user's content receiving device in accordance with the first delivery rule and the determined next portion.

### Brief Description of the Drawings

FIG. 1 shows a flowchart of the electronic publication process of the present invention.

FIG. 2 shows a diagram of the flexibility of the variable schedule publication of the present invention.

FIG. 3 shows a block diagram of a system using the electronic publication process of the present invention.

5 FIG. 4 shows an alternate embodiment of the system in accordance with FIG. 3.

FIG. 5 shows a block diagram of an electronic publisher's server in accordance with the present invention.

10 FIG. 6 shows a block diagram of the publisher-service-subscriber system of the present invention.

#### Detailed Description of the Preferred Embodiment

The present invention enables an electronic publisher to distribute, over the Internet or other networks, content, information or the like having a serial nature (e.g., periodical issues) while accommodating a schedule that is convenient to the consumer. By allowing a consumer to subscribe to a multi-issue publication and set their own schedule for distribution, a service enables the consumer to read the publication at their leisure and allows the electronic publisher to not be required to monitor the content for time sensitive material. Moreover, the publisher often wishes to maintain control over certain parameters of the publication for artistic, economic, timing, and other reasons. The publisher can require that the sequential information (content) be delivered only in ascending sequential order, for example. Or, the sequential information can only be delivered as a portion, a single issue, in the sequence and only be delivered in a particular time frame. For artistic reasons, an author may require that one chapter of a novel must be followed by a period of time during which the next chapter will not be allowed to be delivered. The present invention offers a new physical and temporal dimension to works of authorship while still enabling a consumer to receive the content in portions on a convenient schedule.

The process of a preferred embodiment of the present invention is illustrated in FIG. 1. The process is based on transactions between a consumer (also referred to as the "subscriber" or "user") and a publisher (and the process is known as the

“publication server” or “server process”). For each publication to be delivered in the manner contemplated by the present invention, the publication server must be configured with an issue locator mechanism, the maximum issue number, and the publisher’s prescribed delivery rule.

5           A locator mechanism, as used herein, refers to an addressing entity and any associated algorithm that is required to uniquely identify, locate, and access the contents of a particular web page, file document, or other object. A URL is one type of locator in common use. A Uniform Resource Locator (URL) is a text string that conforms to a well-defined standard syntax and embodies a general addressing  
10       protocol for web pages, email users, files, and other objects that are accessible via the Internet. Another type of locator is the text string used to fully specify the name and location of a file within a computer’s file system.

          Actual issue content is ultimately retrieved in accordance with whatever locator mechanism is employed. In a preferred embodiment, the locator mechanism is  
15       a base URL, and the specific issues of publication content are obtained at the locations addressed by issue-number variants of the base URL.

          A base URL is a template that the server uses to construct the complete locator for all issues of a publication. A locator example might be “http://foo.com/tcq-##-@@.pdf”, where the server process is designed to recognize “##” and “@@” as  
20       markers. The markers are then replaced with, in one embodiment, a 4-digit issue number and a 6-character “random character key”, respectively. The “issue number variant of the base URL” would be the phrase that results from the substitution (e.g., http://foo.com/tcg-0001-B9DXF3.pdf). This locator scheme is discussed in more detail subsequently.

25       A significant portion of the publisher’s delivery rule is a computer-readable description of the publication’s delivery time and frequency, as prescribed by the publisher. Examples of delivery rules might be “every other week at noon on Thursday” (“W2 TH 1200”), or “every day at 6 AM” (“D1 0600”).

          The subscriber first subscribes to an electronic publication service provider  
30       (step 101). Of course, the subscriber employs a receiving device or a computing device such as a personal computer, a personal digital assistant, web appliance, or

similar apparatus that will receive serialized content from the service provider. The subscriber contacts the publication server of the service provider, requesting a new subscription. A server software process initiates the subscription by creating a new, uniquely identifiable subscription record. The subscription record is created in a location (e.g., a database) that can be freely accessed, queried, and modified by the server, and is initialized with at least two key pieces of subscriber parameter information: The time and date that the subscription was created ( $T_0$ ), and the subscriber's personal current issue number ( $N_{\text{subscriber}}$ ). The issue number ( $N_{\text{subscriber}}$ ) is initially set to zero to indicate that the subscriber has not yet received the first issue.

The time and date when the subscriber created the subscription, the subscription inception date, ( $T_0$ ) is needed in order for the publication server to calculate, on any given date, the number of the "publisher's current issue" ( $N_{\text{publisher}}$ ) within the sequence. This is the highest-numbered issue that, according to the publisher's delivery rule, the subscriber is entitled to receive. The subscriber's personal current issue number ( $N_{\text{subscriber}}$ ) is used in the process of the present invention to track where the subscriber is located in the serialized content.

The delivery parameter ( $R_{\text{publisher}}$ ) and the maximum issue number ( $N_{\text{max}}$ ) for the particular publication are now determined (step 102). The delivery parameter effectively defines the maximum frequency at which sequential issues can be released to the subscriber. A copy of the delivery parameter, which is defined and maintained on the service provider's server, is sent to (step 103) and subsequently used by the subscriber's client software as the subscriber's default delivery rule that specifies when to request the next issue of content. In an alternative embodiment, the subscriber can alter their copy of the delivery rule at any time to fit their own schedule. Each sequential issue is, nevertheless, released to the subscriber, however, only in accordance with the issue release schedule prescribed by the delivery parameter.

On a regular basis, according to the subscriber's schedule, the client software attempts to fetch (step 105) the next issue of content by sending an issue fetch request to the server. The server fields the request and invokes a process that examines certain subscription details (step 106). The subscription details include the delivery

parameter, the current date, the inception date, the maximum available issue number, and other portions of the publisher's rule where necessary.

Thus, using the publisher-specified delivery rule, including the delivery parameter, the current date ( $T_{\text{now}}$ ), the sign-up date ( $T_0$ ), and the maximum available issue number ( $N_{\text{max}}$ ), the process at the server calculates the "publisher's current issue" ( $N_{\text{publisher}}$ ). The publisher's current issue is determined by first calculating the number of unique issues that, according to the delivery parameter, would normally be made available between the subscriber's inception date ( $T_0$ ) and the current time and date ( $T_{\text{now}}$ ). The calculated value must not exceed the maximum available issue number ( $N_{\text{max}}$ ).  $N_{\text{publisher}}$  thus represents the maximum issue number that the subscriber is entitled to see, i.e., the entitled issue.

Stated another way, the publisher's current issue number is calculated by the function  $\text{MIN}(N_{\text{max}}, \text{NUM\_ISSUES}(R_{\text{publisher}}, T_{\text{now}}, T_0))$ . In this function, the delivery parameter,  $R_{\text{publisher}}$ , the current time,  $T_{\text{now}}$ , and the time of initial subscription,  $T_0$ , define the function  $\text{NUM\_ISSUES}()$ . This yields a number of unique issues that the publisher would make available through the service provider during a time interval from  $T_0$  to  $T_{\text{now}}$ , and the function  $\text{MIN}()$  returns the lesser of two values.

The server process next compares the publisher's issue number ( $N_{\text{publisher}}$ ) with the subscriber's personal current issue number ( $N_{\text{subscriber}}$ ) (step 108). If the subscriber's personal issue number is less than the publisher's number, or if the subscriber's personal issue number is equal to zero (step 108), the subscriber's personal issue number is incremented (step 110) by one.

If the subscriber's personal issue number is not less than the publisher's issue number, a delivery error condition is generated (step 109) and sent to the subscriber. The error is an indication that a next issue is not available. The subscriber has attempted to fetch an issue that, according to the publisher's delivery schedule, the subscriber is not yet entitled to receive, or, according to the maximum available issue number, does not exist.

If the subscriber's issue number ( $N_{\text{subscriber}}$ ) was successfully incremented, the incremented value is used to construct a locator (e.g., a URL or filename) (step 111) for the corresponding issue of content, and the issue is then delivered to the subscriber

(step 115). In the preferred embodiment, the issue locator is transmitted to the subscriber's computer that may then use the locator to retrieve, via a transfer protocol such as "http" or "ftp", the actual issue content for printing, monitor presentation, or storage.

5 In an alternate embodiment, the server first uses the locator to retrieve the issue contents then sends the actual content to the subscriber for printing, presentation, or storage. The process then waits (step 120) for the next delivery request from the subscriber. This request can be a fetch command, a print on demand command, or a view command.

10 FIG. 2 illustrates an example of a preferred embodiment of the issues or portions of content. Individual portions of the content exist as randomly selectable files in a common repository where each file's name identifies its issue number. The subscriber's client software accesses the content by performing a file transfer protocol (FTP) from a URL that is obtained from the server.

15 Initially, the publisher may store already completed issues of content (200) at the URL even though the entire sequence is not yet completed. The publisher can then store additional portions of content in the future (210) when it has been completed.

The content may be stored at the same service provider server as the one discussed above or a different server. In an alternate embodiment, the content is spread among multiple servers. The publication server process that fields issue retrieval requests uses an algorithm or mechanism to construct, given a specific issue number, the appropriate locator needed to retrieve the actual issue content.

20 In a preferred embodiment, the subscriber's personal issue number is used to construct the locator for the issue to be delivered. In an alternate embodiment, the issue number and other constant publication-unique information may be used to generate a seemingly random character key to obfuscate the filename, making it almost impossible for a person who knows the URLs of the already-delivered content to deduce the URLs of upcoming issues and look ahead at yet-undelivered, postponed, and/or unpaid for, content.

30 One method for creating a suitably obfuscated name is to assemble each issue's filename from (1) an unchanging name portion, concatenated with (2) a fixed

number of digits representing the specific issue number. As an example, issues 1 through 3 of "The Chicago Gazette" might be named with a unique signature like "tcg-0001-RYSLIE.pdf", "tcg-00020-C34LLW.pdf", and "tcg-0003-HQ9LL6.pdf".

In this example, the initial label portion ("tcg") uniquely distinguishes files belonging to "The Chicago Gazette" from other unrelated publications that may reside in the same repository. The label and issue number ("tcg-0001"), taken together, uniquely identify the file for a specific issue. This label/number pairing is attractive in that it facilitates generation of a well-organized list of all files in the repository. An alphanumeric sort of the names of everything in the repository will group together in order, by issue number, all of the files belonging to each publication.

A text-scrambling algorithm generates the "random character" portion. Given the label/number information, the algorithm produces a seemingly random sequence of valid filename characters (e.g., "tcg-0001" yields "RYSLIE"). A corresponding decryption algorithm is not needed. The characters serve only to obfuscate the final name and thereby make it improbable for anyone who does not know the scrambling algorithm to determine the complete filename for a specific issue.

The final portion of the name (".pdf") in this example is well known in the art as the filename extension that identifies the file type. A ".pdf" file generally embodies a document in Adobe Portable Document Format.

The above description of the generation of the locator mechanism is only one embodiment. The present invention encompasses other methods for generating a locator mechanism.

If the subscriber has already received the calculated issue, the content is sent only if the subscriber has elected to have content delivered even if it has not changed. If the subscriber elects to not re-receive content that has already been once delivered, this mechanism effectively suspends delivery when all available issues have been delivered.

If the subscriber does not cancel the subscription, the client software continues to regularly check for new content. All that is necessary to resume delivery is for the publisher to put (or the service provider to obtain) new, appropriately named content into the repository on the server and adjust the maximum available issue number.



The publisher-specified schedule determines the maximum rate at which new issues are released to the subscriber. This is analogous to the periodical frequency of issue for a typical dated publication. If the subscriber adjusts the schedule to fetch less frequently than the publisher's default schedule, a new issue will always be available at each fetch. If the user adjusts the schedule to fetch more frequently, some delivery attempts will determine that the content has not yet changed when, at the time of the request, the calculated publisher's current issue number ( $N_{\text{publisher}}$ ) does not yet exceed the value of the subscriber's personal current issue number ( $N_{\text{subscriber}}$ ). In this case, the issue will not be printed or viewed. In an alternate embodiment, the client may be configured to print or view old content.

If the subscriber turns off his computer and/or client for an extended period of time, no issues are lost since the subscriber's personal current issue number ( $N_{\text{subscriber}}$ ) advances only in response to a delivery request and, therefore, remains unchanged during the downtime. The publisher's current issue number ( $N_{\text{publisher}}$ ) meanwhile continues to advance, since it is a computed value based on elapsed time. When the client resumes operation, deliveries resume where they left off.

In a preferred embodiment, the client is responsible for storing the subscriber's preferred delivery. In an alternate embodiment, this schedule is stored on the service provider's server and the server is responsible for transmitting the issue to the subscriber's client software.

If the subscriber's personal, or last delivered, issue number ever lags the publisher's issue number, such as when the subscriber's client is turned off or the subscriber's delivery schedule is substantially slower than the publisher's schedule, the subscriber's most recently printed issue will be one or more issues behind the publisher's current issue. In this case, the subscriber can catch up by repeatedly requesting issue delivery (e.g., by using a "Print Now" feature of the client software) to quickly fetch the additional sequential issues to which they are entitled, but not go beyond the issue that the service provider's server determines to be the publisher's current installment, i.e., the issue number for entitled issues. Alternatively, the subscriber can request catch-up deliveries of content that run from the last received issue number to the maximum issue number for entitled issues available to the

subscriber. The entitled issues, of course, cannot exceed the publisher's maximum issue number and must be compliant with any other portions of the publisher's delivery rule that might, for example, place a limit on the maximum number of issues that can be delivered in a single delivery. Also, the subscriber can request back issues  
5 from the service provider. A determination is made that the back issue has an issue number smaller than the subscriber's most recently received, or current, issue and a check is made that this back issue number is less than the maximum issue number for entitled issues and otherwise is compliant with any other portions of the publisher's delivery rule before allowing delivery of the back issue.

10 FIG. 3 illustrates a preferred embodiment system of the present invention. This system is comprised of a computer device (301) that is coupled to a network interface device (305), such as a modem. The interface (305), which may be either external to or incorporated within the subscriber's computer device (the content receiving device) (301), facilitates transfer of information, both reception and transmission, between the  
15 computer device (301) and the Internet (310) via telephone lines, network cable, radio frequency emission, or other medium for which the interface device (305) is designed.

The subscriber's computer device employed in a preferred embodiment of the present invention, is one that comprises a memory, a keyboard, a monitor, a nonvolatile storage device, and a central processor. Such a computer is one that runs  
20 operating systems such as WINDOWS and MACINTOSH and is well known in the art.

The service provider's server (315) is also coupled to the Internet through a network interface device (320). This interface device (320) can be either a stand-alone item or be incorporated in the circuitry of the server (315).

25 In alternate embodiments, the subscriber's computer device (the content receiving device) takes the form of a personal digital assistant such as a PALM device, HANDSPRING VISOR, or APPLE NEWTON. An example of such a personal digital assistant is illustrated in FIG. 4.

FIG. 4 shows a personal digital assistant (401) coupled to the Internet (410)  
30 through either a built-in modem or a stand-alone modem coupled between the personal digital assistant and the telephone lines. The service provider's server (415)

is also coupled to the Internet through a built-in or stand-alone network interface device.

In still other alternate embodiments, the subscriber's computer device takes the form of a wireless access protocol enabled wireless telephone or an electronic  
5 book that is capable of downloading novels and other information over the Internet. Any computer-based device that can send and receive information for the subscriber over the Internet is encompassed by the present invention.

The content receiving device (computer or personal digital assistant) of the present invention runs the client process of the present invention. In a preferred  
10 embodiment, the client process is a software application that is resident in the computer or personal digital assistant. The client process is responsible for providing the handshaking with the service provider's server. The handshaking process includes the subscriber logging in with his or her account information and/or a secret password.

The client process also tracks the downloading over the Internet of the issue  
15 content from the service provider's server. Additionally, the client process is responsible for providing the subscriber's subscription identification number to the service provider's server during the handshake process.

The client process provides the user interface for the subscriber to view and print the content that has been downloaded. The user interface includes the controls  
20 and drivers required to move about in the content and print the content to a local or network printer.

An alternate embodiment of the client software is a Web browser such as NETSCAPE NAVIGATOR or MICROSOFT INTERNET EXPLORER. The subscriber can type in all of the required information to perform the handshake with  
25 the publisher's server to log on. For example, the subscriber can type in his account number, password, and select the document desired.

The subscriber's personal issue number is stored at the service provider's server. The subscriber's unique subscription identification number is stored at the subscriber's computer.

30 FIG. 5 illustrates a block diagram of typical computer network server device used by a service provider. This block diagram may also illustrate a computer device

used by the subscriber. However, if a personal digital assistant or electronic book is used in the present invention, the block diagram of the computer device may not have all of the elements illustrated in FIG. 5. Additionally, a computer device does not require the removable media in order for the present invention to operate properly.

5       The server/computer device is comprised of a central processing unit (CPU) (501) that controls the device. The CPU (501) is coupled to the memory (505) over an address bus, data bus, and control bus. In the preferred embodiment, the memory (505) includes both read only memory (ROM) for storing data permanently and random access memory (RAM) for storing data temporarily.

10       The device is further comprised of a nonvolatile storage device, such as a hard drive (510), for storing software applications, such as the client process, to be accessed by the CPU (501) and executed in memory (505). A keyboard and monitor (515) are used to enter data and display data, respectively. The modem (520) provides the interface between the computer and the telephone lines. A removable media drive (525), in the preferred embodiment, is a floppy disk drive. Alternate embodiments can have one or more of the following as a removable media drive: CDRom drive, DVD ROM drive, Flash Memory card, Magnetic Tape drive, and ZIP drive.

15       The delivery process of the present invention may or may not be fee based. If the publisher or service provider does not charge a fee to subscribe to the delivery process, the publisher or service provider may include advertisements along with the current issue downloaded by the subscriber. The advertisement may be geared toward a special event that is to occur on the next scheduled date for delivery. For example, if the subscriber is expected to read the next issue of content around April 15, the advertisement may be directed towards products such as tax software.

20       Additionally, when the subscriber retrieves the last installment of the sequential issues of content, the publisher may include a "suggested readings" section. This section informs the subscriber of additional products offered by the publisher that may be of interest to the subscriber since they are similar to the just-read issues.

25       FIG. 6 illustrates an example of the publisher/server/subscriber system of the present invention. A content repository (601) is designed and located such that the service provider can freely place new issues of information either directly or

30

indirectly, via the publication server (605) or other intermediary system, into the repository (601). The server (605) must be able to construct a suitable locator (e.g., a URL or filename) that the server, subscriber client, browser, or other subscription delivery process can then use to retrieve a specific issue of information from the group  
5 of available issues that exist in the repository (601). In the preferred embodiment, the repository (601) exists at the server (605) that is administered by the service provider, which obtains content from a publisher.

In an alternate embodiment, the repository (601) exists in a location maintained by the publisher and made accessible to other processes that include one  
10 or more service provider servers or subscriber clients. In another embodiment, the server (605) is also administered by the publisher.

The publisher, in this example, prepares a group of five "weekday" issues of information (630) that is obtained for deposit in the content repository (601) on Sunday in preparation for the upcoming week. The publisher also prepares a group of  
15 two "weekend" issues of information (631) that is obtained for deposit in the content repository (601) on Friday in preparation for the upcoming Sunday. Individual issues of information are sequentially and uniquely numbered in order of their intended release. If, in this example, the publisher's initial group of issues is the weekday block (630), the issues for Monday through Friday would be numbered 1 through 5,  
20 respectively. The weekend block (631) would then contain issues 6 and 7, both for release on Sunday. Continuing in this manner, a subsequent group of weekday issues for the next week would begin with issue 8.

In an alternate embodiment, there is only one group of multiple issues of information that the publisher transfers into the repository (601) on a regularly  
25 scheduled basis. In another embodiment, there are multiple groups of information that the publisher transfers into the repository (601), each group having a unique schedule of deposition that is set by the publisher.

The service provider's server (605) has an appropriate means of determining, for any specified time and date after the creation of a new subscription, the number of  
30 the specific issue of information that the subscriber (650) is entitled to receive in

accordance with an issue release schedule. In the preferred embodiment, the release schedule is prescribed by the publisher and agreed to by the issue publication service.

In an alternate embodiment, the release schedule is prescribed by the issue publication service and agreed to by the publisher. In this example, the publisher's  
5 Delivery Schedule stipulates that one issue shall be released each day on Monday through Friday, there shall be no issue released on Saturday, and there shall be two issues released on Sunday.

Once the server (605) is prepared with the mechanisms for determining the number of the issue that the subscriber is entitled to receive, for determining the  
10 locator required in order to retrieve the issue content, and is aware of the maximum issue number for the issued of information located in the repository (601), the server (605) is ready to field requests from subscriber clients to initiate subscriptions and deliver issues. Once the subscriber client (650) is prepared with an issue delivery schedule, it is ready to request delivery of issues of information from the server (605).  
15 In a preferred embodiment, the client's issue delivery schedule is the same as the publisher's issue release schedule. In an alternate embodiment, the client's issue delivery schedule is defined by the subscriber.

FIG. 6 illustrates the subscriber (650) retrieving the information in three different groups (640 – 642) in accordance with a custom issue delivery schedule that  
20 has been defined by the subscriber (650). The subscriber (650) is requesting two deliveries on Tuesday (640), two deliveries on Thursday (641), and three deliveries on Sunday (642).

On Sunday, the publisher deposits five issues (606 – 610) to be released Monday through Friday on a daily basis. The subscriber's two delivery requests on  
25 Tuesday (640), therefore, are allowed to retrieve the Monday (615) and Tuesday (616) issues of information. Similarly, the subscriber's two delivery requests on Thursday (641) are allowed to retrieve the Wednesday (617) and Thursday (618) issues of information. On Friday, the publisher deposits the two weekend issues (611 – 612), both of which are to be released on Sunday. The subscriber's next group of delivery  
30 requests occurs on Sunday (642) and is allowed to retrieve the Friday (619) and both Sunday (620 – 621) issues of information.

The example of FIG. 6 is for illustration purposes only. The present invention is not limited to the groupings and scheduling illustrated in this figure.

In summary, the present invention provides a publisher with the ability to publish material once without the need to periodically update or monitor it. Users can  
5 subscribe to a service provider's offering of the content, paying for the subscription, for many years after publication. The delivery of all issues is guaranteed since each subscription, regardless of when the subscription is initiated, begins with a specific first issue, and is delivered according to sequential issue numbers rather than relying on the customer to fetch the content on a specific date. Batches of publications such as  
10 crossword puzzles and quotes of the day can be deposited in a single location and, by using the issue number and publication-specific constants to generate and embed a seemingly random key into the URL, it is nearly impossible for one to deduce, through examination of previously exposed URLs, the URL for an upcoming issue and look ahead to see future content.

15 Benefits for the subscriber include the flexible downloading of the issues of content. The present invention allows one subscriber to read a novel over a six week period while another subscriber can take six months to finish it. Additionally, the subscriber's subscriptions do not stack up if they are not downloaded right away. They can be suspended and resumed at any time. Moreover, the publisher can  
20 maintain a supervision over the sequence and timing of delivery long after the initial offering of the content.

What is claimed is: